Docket No.: M&N-IT-466

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date indicated below.

By: _____ Date: July 23, 2003

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Eric Liau
Applic. No. : 10/623,067
Filed : July 18, 2003
Title : Method of Generating a Test Pattern for Simulating and/or Testing the Layout of an Integrated Circuit

CLAIM FOR PRIORITY

Commissioner for Patents,
P.O. Box 1450, Alexandria, VA 22313-1450

Sir:

Claim is hereby made for a right of priority under Title 35, U.S. Code, Section 119, based upon the European Patent Application 020 90 271.4, filed July 19, 2002.

A certified copy of the above-mentioned foreign patent application is being submitted herewith.

Respectfully submitted,

_____
For Applicant

MARKUS NOLFF
REG. NO. 37,006

Date: August 14, 2003

Lerner and Greenberg, P.A.
Post Office Box 2480
Hollywood, FL 33022-2480
Tel: (954) 925-1100
Fax: (954) 925-1101

/av

Europäisches Patentamt	European Patent Office	Office eur péen des brevets

Bescheinigung | Certificate | Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

02090271.4

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Anmeldung Nr:
Application no.: 02090271.4
Demande no:

Anmeldetag:
Date of filing: 19.07.02
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Infineon Technologies AG
St.-Martin-Strasse 53
81669 München
ALLEMAGNE

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Method of generating a test pattern for the simulation and/or test of the layout of an integrated circuit

In Anspruch genommene Priorität(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

G01R31/00

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR IE IT LI LU MC NL PT SE SK TR

Description

Method of generating a test pattern for the simulation and/or test of the layout of an integrated circuit

To achieve high perfomance and high integration density, the dimensions of integrated circuit components are scaled down more and more. In particular, transistor dimensions are scaled down while lower power dissipation is achieved by scaling down the supply voltage. However, due to high packing density of transistors, the power supply current is increasing, and hence, large current swings within a short period of time can cause considerable noise. As a consequence, one difficulty circuit designers face is the power delivery of very high performance circuits due to the severe switching noise.

In order to verify the function of a newly designed integrated circuit, the circuit is first simulated and then tested. During simulation, multiple input signals are applied to the inputs of the circuit, and the output signals of the circuit calculated. The input signals are referred to as test patterns. If the output signals do not sufficiently approximate preset target signals, the circuit is redesigned and resimulated.

Subsequently, when simulation is completed, a chip containing the integrated circuit is manufactured and tested using ATE (Automatic Test Equipment). The ATE also applies a test pattern to the circuit. The test pattern for the ATE has to be input manually by a user. Generally, the same test pattern that has been used for simulation is also used for testing. If the output signals generated by the circuit in response to the test pattern of the ATE deviate from preset target signals, the circuit is redesigned, resimulated and retested.

As the complexity of integrated circuits increases, integration density and functionality increases dramatically. The simultaneous switching of a large number of transistors induces a large current spike. The switching noise on the power distribution network must be suppressed to a tolerable level to ensure the reliability of the circuit. In order to efficiently combat the switching noise, estimation of the worst case switching noise is required.

On way of determining the worst case switching noise is to simulate all combinations of input patterns to determine which combination will induce the maximum switching noise. However, the complexity of the solution space is exponentially proportional to the number of primary inputs of the system. Accordingly, it would require an enormous time to process the entire solution space for even a moderately complex system.

To this end, a number of approaches have been proposed to deal with these problems. In "Estimation of Switching Noise on Power Supply Lines in Deep Sub-micron CMOS circuits", Shiyou Zhao and Kaushik Roy, 13th International Conference on VLSI Design, IEEE January 2000, there is proposed a probabalistic approach to determine the lower bound of the worst case switching noise on power supply lines. The algorithm described therein traces the worst case input patterns which induces the steepest maximum switching current spike and therefore the maximum switching noise. This is based on the observation that the maximum switching noise is directly related to the steepest maximum switching current spike.

In this approach, the design of an integrated circuit is simulated by applying randomly generated input signal vectors to the inputs of the circuit. For each input vector pair, the simulated peak switching current is determined. The worst case input vector pairs feed, as initial population, a

genetic algorithm. The genetic algorithm is designed to single out the near optimal input pattern(s) that induce the steepest maximum switching current spike and, therefore, the worst case switching noise. The worst case input patterns are then used in HSPICE simulation of the circuits to extract the exact current waveform.

One problem associated with this approach is the difficulty of generating suitable random test patterns. The larger the number of random test patterns, the higher the likelihood of generating a test pattern which approximates the worst case sufficiently. However, since the simulation of each test pattern is time consuming, the simulation of a large number number of test patterns is not practical.

In particular, if a genetic algorithm is used, it is too time consuming to simulate every single random pattern out of every new pattern population before the algorithm is able to determine which of the patterns of the population is to be selected for further optimization. Therefore, this method becomes saturated by the number of trial random patterns in each pattern population. It is suitable for small circuits. However, it could take up to years to perform a full chip simulation of a large circuit using even the fastest simulation applications.

The present invention aims to address this problem.

According to one aspect of the invention, there is provided a method of generating a set of test patterns for the simulation and/or test of the layout of an integrated circuit, the method comprising the steps of:

(a) generating a set of test patterns on a random basis;

(b) applying the set of test patterns to the integrated circuit using automatic test equipment (ATE);

(c) determining the outputs of the integrated circuit;

(d) processing the outputs to determine whether predetermined test criteria are met;

(e) depending on the determination in step (d), generating a new set of test patterns on the basis of the set of test patterns generated in step (a) using a genetic algorithm.

Accordingly, the method employs a genetic algorithm (optimization method) to optimize a set of random patterns based on measurements using an ATE. Thereby, a set of worst case noise patterns can be selected automatically.

The selected set of worst case noise patterns can be used to re-simulate the layout of the integrated circuit for detail design analysis or improvement.

This approach is more efficient and faster than any pure random-based method employing a non-optimal random pattern generation or simulation approach.

In addition, it can be used with any ATE system.

A genetic algorithm can be equally employed to detect a set of potential worst case test conditions in respect of input signals supplied to an integrated circuit (e.g. power supply). The parameters of such input signals are referred to as AC/DC parameters.

Accordingly, there is also provided a method of generating a set of input signals for the simulation and/or test of the layout of an integrated circuit, the method comprising the steps of:

(m) generating a plurality of sets of input signals;

(n) applying the plurality of sets of input signals to the integrated circuit using automatic test equipment (ATE);

(o) determining the outputs of the integrated circuit;

(p) processing the outputs to determine whether predetermined test criteria are met;

(q) depending on the determination in step (p), generating a new plurality of sets of input signals on the basis of the plurality of sets of input signals generated in step (m) using a genetic algorithm.

According to this method, a set of worst case test conditions due to resulting high power supply noise (dynamic current) can be detected without having to manually and/or randomly select parameters of input signals.

This method may also be implemented using existing ATE systems.

The input signals obtained as a result of this method may subsequently be re-simulated.

Preferably, the method of approximating a worst case set of test pattern is performed together with the method of approximating a set of worst case input signals parameters. This way, an overall worst case condition of operation of the integrated circuit can be approximated.

Exemplary embodiments of the invention will now be described by reference to the drawings, of which

Figure 1 illustrates a schematic flow diagram of a method according to an embodiment of the invention;

Figure 2 illustrates the results of a practical experiment using a method according to an embodiment of the invention; and

Figure 3 illustrates a schematic flow diagram of a method according to another embodiment of the invention.

Genetic algorithms are based on the principles of natural selection. In particular, genetic algorithms are stochastic

search methods which simulate natural biological evolution. The algorithms operate on the basis of a population of potential solutions and, applying the principle of "survival of the fittest" to these potential solutions, produce a better approximation of a target solution in each iteration of the algorithm.

Each iteration of the algorithm produces a new generation of approximations. The approximations of each generations are created by the process of selecting individuals according to their level of "fitness" in the problem domain. The selected individuals are bred with one another using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited for their environment than the individuals from which they were created, just as in natural adaptation.

Accordingly, genetic algorithms model natural processes such as selection, cross over, recombination and mutation.

Figure 1 shows a method for detecting the worst case current consumption/peak current pattern (RSMA) based on a genetic algorithm. This method operates on the basis of populations of individual patterns instead of a single pattern solution. In this way, the search for better approximations can be performed in a parallel manner. Therefore, this method is more efficient than single pattern searching processes using dynamic random algorithm methods.

Genetic algorithms may be employed for the simulation of an integrated circuit design in order to solve the worst case pattern search problem. The efficiency of genetic searching procedures is largely dependent on the number of pattern populations and the number of test patterns in each pattern population. However, as indicated above, the simulation-based approach forms a limitation if genetic algorithms are to be employed. The genetic selection procedure has to

evaluate every "fitness" (dynamic peak/averaged current) of the test patterns in each pattern population. For example, there may be 200 pattern populations each including 20 patterns. Thus, the genetic algorithm has to evaluate the fitness of 200 * 20 = 4000 patterns. If each test pattern is a 50 cycles test pattern which requires 30 minutes of simulation time (e.g. EPIC oder SPICE simulator), then the total required searching and simulation time is 4000 * 30 minutes = 120000 minutes, i.e. approximately 83 days of non stop simulation in order to process 200 pattern populations only.

In addition, the full pattern combination domain increases proportionally to the complexity of VLSI or ULSI designs. Therefore, a subset of 200 pattern populations is only a very small subset of the full pattern combination domain.

In contrast, when using a genetic algorithm together with ATE, many more pattern populations per time unit can be processed. This is because the testing of an integrated circuit using ATE is considerably faster than simulation using conventional systems. Accordingly, the approximation of worst case test patterns in a given period of time is much more accurate.

An implementation of a dynamic genetic algorithm for use with ATE is presented in the following. At the beginning of the computation, a number of individual random patterns

$$P_N^{POP} = (p_1, p_2, \ldots, p_N)$$

are randomly generated and initialized, wherein N is the maximum number of random patterns and POP is the maximum number of pattern populations.

Subsequently, for each individual pattern $(p_1, p_2, \ldots, p_N)$, the objective functions

$$I_{peak}(\forall\ I_{sample}(P_N,\ SRMS))$$

and

$$I_{averaged}(P_N,\ SRMS)$$

are evaluated using equation (1):

$$I_{Measurement}(P_N,T)=\frac{V_{DD}(P_N,T)}{R_{eff}}+\frac{1}{L_{eff}}\int_{T_{min}}^{T_{max}}V_{DD}(P_N,T)dT+\Delta I_{CMOS}(P_N,T),\ \forall T,P_N>0$$
$$T=SRMS(T_{min},T_{max})\Rightarrow Random\_Float\_Number(T_{min},T_{max}) \quad (1)$$
$$T_{max}\geq T_{min},\ \forall T_{min},T_{max}>0,\ I_{Measurement}(P_N,T)\in\{I_{peak}\quad I_{averaged}\}$$

The first (initial) generation is thus produced, and the averaged fitness of the individual patterns (p_1, p_2, ..., p_N) is calculated using equation (2):

$$Averaged\_Fixness(Fixness(P_N^{POP}))=\frac{\sum_{N=0}^{N}I_{Measurement}(P_N)}{N},\ N,P_N>0 \quad (2)$$
$$Fixness(P_N)=I_{Measurement}(P_N,T)\in\{I_{peak}\quad I_{averaged}\}$$

If the optimization criteria

$$(Averaged\_Fitness(I_{Measurement}(P_N^{POP}))\ <\ I_{MAX\_REF})$$

is not met for any existing population, a new population is created on the basis of the existing population. Individual patterns are selected according to their fitness for the production of offspring (loop1 in Figure 1).

In this selection approach, the basic concept of tournament selection is employed. That is, only the best individual pattern from the existing population is selected as a parent. This process is repeated until a pre-defined percentage of best patterns has been selected:

$$Sorting(I_{Measurement}(P_N) \in \{I_{\min}(P_{N_{\min}}) \quad \ldots\ldots \quad I_{\max}(P_{N_{\max}})\}) \Rightarrow Parent(I_{Measurement}(P_N))$$
$$N \in \{N_{\min} \quad N_{\max}\} \qquad N \in \{N_{\min} = N_{\max} - (N_{\max} \times B) \quad N_{\max}$$

(3)

wherein B is the pre-defined percentage of the best pattern group. The sorting function first re-arranges the test patterns from minimum to maximum according to their fitness values. Subsequently, the parent selection is generated in random sequence based on the new sub-optimal fitness range N, which is calculated using B. Parents (selected patterns) are combined using cross over (4), re-combined (5) and mutated (6) in order to produce offspring:

$$CrossOver(P_N(C_1,C_2),P_{N+1}(C_3,C_4)) \Rightarrow Upper\_CrossOver(P_N(C_3,C_2),P_{N+1}(C_1,C_4))$$
$$\Rightarrow Lower\_CrossOver(P_N(C_1,C_4),P_{N+1}(C_3,C_2)) \qquad (4)$$
$$\Rightarrow Stripe\_CrossOver(P_N(C_4,C_3),P_{N+1}(C_2,C_1))$$

where C is the test pattern content which is selected for cross over of two patterns. In the cross over process, upper, lower or stripe cross over methods are performed in random sequence, and the contents of two cross over patterns are exchanged in order to produce two new offspring patterns. Thereafter, the re-combination equation (5) is used to select the best fitness pattern out of two new cross over offspring patterns:

$$Recombination(P_N,P_{N+1}) \Rightarrow I_{maximum}(P_N,P_{N+1}) \Rightarrow I_{Best}(P_M),\ N,M,P_N > 0 \qquad (5)$$

$$Mutation(P_M(C_1,C_2,C_3,C_4....C_y)) \Rightarrow P_M(C_1+R_1,C_2+R_2,.....C_y+R_y)$$
$$R_y \in \{1 \quad 0 \quad -1\},\ M,P_M,y > 0 \qquad (6)$$

where M is the number of new selected offspring patterns to form the new population. After recombination, the offspring

undergoes mutation. Offspring variables are mutated by the addition of small random values

$$Ry \in \{1 \quad 0 \quad -1\}.$$

The mutation process helps to improve the optimization search process.

Finally, all offspring patterns are inserted into the population, replacing the parents (original pattern population) and producing a new generation. This cycle (loop 1 in Figure 1) is performed until the optimiziation criteria are met.

If the fitness does not improve after a pre-defined number of genetic breeding generations, a new pattern population (loop 2 in Figure 1) will be generated in random sequence. This combination greatly increases the chances of finding worst case test patterns.

A complete implementation of this algorithm using ATE J973 is given in Annex 1.

Figure 2 illustrates an experimental plot of a dynamic genetic algorithm approach (100MHz, VDDP IO Pad eDRAM core). The experiment included 20 patterns in each population and 34 (population) genetic algorithm search samples (loop 2 + loop 1). The total number of patterns processed was 20 * 34 = 680. The experiment lasted considerably shorter than a simulation-based approach, and the current bound obtained through using the worst case pattern was higher compared to conventional dynamic random algorithm-based approaches.

Another embodiment of the invention will now be described with reference to Figure 3.

As indicated above, the maximum switching noise occuring during operation of an integrated circuit is directly related to the steepest maximum switching dynamic current spike. The switching behaviour depends on the test patterns applied to the circuit.

In addition, the switching noise depends on the deviation of input signals (AC & DC conditions/parameters) applied to the integrated circuit. This reflects operation of integrated circuits in practice, where a 10% power supply fluctuation can occur in most applications.

Accordingly, signal switching and deviation can have a great impact on internal signal behaviour (e.g. propagation delay) and switching noise. However, using the simulation-based approach, it is not possible to simulate millions of test patterns and different input signal conditions at the same time. Therefore, simulation-based approaches take into account signal switching only and ignore the other important cause of switching noise, namely signal deviation.

According to the embodiment of Figure 3, a method is provided that allows for signal switching and signal deviation analysis at the same time. This results in a more efficient analysis of the operational behaviour of an integrated circuit than known methods.

Worst case conditions of operation of an integrated circuit depends the level of switching noise. The switching noise is mainly influenced by signal switching events (different test patterns) and signal deviation (AC/DC parameter variation). In order to approximate a worst case condition, the steepest maximum switching dynamic current is to be detected. The switching dynamic current is a function of the test pattern (signal switching) and test condition variation (signal deviation). This can be subject to the constraint that the test patterns have to be valid. Alternatively, the

approximation can be implemented without constraint such that the dynamic current function explores the fail region without any input signal deviation limits (e.g. +/-12.5%).

An implementation of an approximation of worst case signal switching and deviation using a constraint optimization approach is illustrated in Figure 3. Initially, a number of individual worst case random patterns $P_N^{PPOP} = (p_1, p_2, \ldots, p_N)$ are generated using the above described method, wherein N is the maximum number of worst case patterns and PPOP is the worst case pre-selected pattern population.

Subsequently, a number of AC or DC parameters $S_M^{SPOP}(AC|DC) = (S_1, S_2, \ldots, S_M)$ are pre-selected, wherein M is the maximum number of worst case conditions and SPOP is a pre-selected worst case condition population. Every single condition population contains a number of pre-selected AC or DC parameters. Multiple populations form a condition code matrix (CCM) file.

Similarly, worst case patterns are stored in a vector code matrix (VCM).

The function for approximating the highest switching noise is defined as a dynamic current function in respect of the switching pattern populations and the condition parameter populations subject to the constraint that the test patterns have to be valid. Thus, the signal switching and deviation analysis is always inside the valid range. It may be useful, however, to detect the boundaries of the valid range.

The power supply switching and deviation noise estimation problem can be formulated as follows:

$$Maximize(I_{switching/\,deviation\_noise}(P_N^{PPOP}, S_M^{SPOP}(AC|DC))),\ \forall N, M, PPOP, SPOP > 0$$

Subject to the constraint: $Pattern\_Pass(P_N^{PPOP})$

$$(AC|DC) \in \{X_{AC\_SPEC} \quad X_{DC\_SPEC}\}_{deviation},\ 0 \le deviation \le 100$$

(7)

As indicated above, evolutionary genetic methods are stochastic optimiziation concepts that mimic biological evolution. Such genetic concept can be employed to detect worst case switching patterns, as described above. The algorithm detects the best fit to the problem by evolution, i.e. processing series of test patterns in parallel. The test patterns are contained in test pattern populations. The fitness is defined as dynamic current in respect of each trial test pattern (signal switching).

Equation (7) reflects test pattern and test condition deviation. In order to produce valid results from input signal deviation with respect to the test patterns, the function must be subjected to the constraint that the test patterns must be valid, since signal fluctuation could otherwise cause test pattern failure.

The fitness in this embodiment is associated with the steepest maximum switching dynamic current, subject to the above mentioned constraint.

Referring more specifically to Figure 3, initially, a number of pre-selected individual test condition combinations (the population/CCM) with respect to a set of worst case patterns (VCM) are randomly generated and intialised. Equation (7) is then evaluated in respect of "fitness" for each combination using ATE, and the averaged fitness of the initial population is calculated.

If the optimization criteria is not met (see "check evolution fitness" in Figure 3), the creation of a new population is initiated. Individual test condition combinations are selected according to their fitness for the production of offspring (Evolution Genetic Optimization Loop). In the selection approach, the basic concept of tournament selection

is employed, i.e. only the best individual combination from this population is selected as parents.

This process is repeated until a pre-defined percentage of best combinations has been selected. The sorting function first re-arranges the test condition combination according to their fitness values from minimum to maximum. Subsequently, the parents are selected in random sequence based on the new sub-optimal fitness range. Parents (selected test condition combinations) are subjected to cross-over, recombination and/or mutation to produce offspring.

In the cross-over process, a method of binary cross-over is used in random sequence, and the contents of two cross-over test condition combinations are exchanged accordingly to produce new offspring combinations. Subsequently, the re-combination process is used to select the best fitness combination out of two new cross-over offspring combinations.

After re-combination, the offspring undergoes mutation. That is, the offspring variables are mutated by the addition of small random values. The mutation rate ranges from -1 to 1. The mutation process helps to improve the optimization search process.

Finally, all offspring are inserted into the population and thus replace the parents (original CCM population), producing a new population.

This cycle is performed until the optimization criteria are met or the end of the optimization loop is reached. If the fitness has not improved after a pre-defined number of genetic breeding generations, a new population is generated in random sequence.

At the end of the evolution loop as shown in Figure 3, the worst case fitness is detected automatically from all

processed populations. The CCM database is updated accordingly.

The whole process is repeated until all worst case patterns from the VCM file have been processed. The final worst case CCM is then generated automatically.

The above described evolution method greatly improves the chances of finding the worst case signal deviation and signal switching combination. At the same time, it is faster and more efficient than known random- or trial-based approaches.

It is to be noted that the invention is not restricted to the embodiments and implementations described herein but encompasses modifications and variations within the scope of the invention as determined from the claims.

Annex 1: Dynamic Genetic Algorithm (D_GA) Implementation Using J973 ATE

Start D_GA: Circuit Initialization

Default AC/DC Specification Initialization.
DP Dummy Pattern : Vector Memory Initialization

$$INPUT: \{N \quad Vector\_Cycles \quad DP \quad Loop1 \quad Loop2 \quad I_{Max\_REF}\}$$

Check if Input valid?
else Input Error! exit(1).

For $POP = 0, 1, 2, 3, \ldots\ldots\ldots, Loop2+1$ do :
{

$$Random\_Pattern\_Generation \Rightarrow P \in (p_1, p_2, \ldots, p_N)$$
$$\forall vector\_cycle, N > 0$$
$$P_N^{POP} = (p_1, p_2, \ldots, p_N) \text{ Initial Pattern Population}$$

For P1 = 0, 1, 2, 3, ..., $N+1$ do :
{

$$Vector\_Code\_Matrix(P_N(Vector\_Cycles)) \Rightarrow \begin{bmatrix} P_0(Vector\_Cycles) \\ P_1(Vector\_Cycles) \\ \vdots \\ P_N(Vector\_Cycles) \end{bmatrix}$$

$$P_N(Vector\_Cycles) \in P_N(vector\_encode(\forall signal\_bus), Vector\_Cycles))$$

$$Pattern\_Generator(Vector\_Memory(P_N))$$
$$\Rightarrow Pattern\_Controller(Vector\_Memory(P_N)) \quad N > 0 \text{ (Pattern Executor)}$$

Start Pattern Generator: $P_N(T_{min}, T_{max}) \Rightarrow Dynamic\_Pattern$
Start Current Measurement & Calculation:

$$I_{Measurement}(P_N, T) = \frac{V_{DD}(P_N, T)}{R_{eff}} + \frac{1}{L_{eff}} \int_{T_{min}}^{T_{max}} V_{DD}(P_N, T)dT + \Delta I_{CMOS}(P_N, T), \ \forall T, P_N > 0$$
$$T = SRMS(T_{min}, T_{max}) \Rightarrow Random\_Float\_Number(T_{min}, T_{max})$$
$$T_{max} \geq T_{min}, \ \forall T_{min}, T_{max} > 0$$

Stop Pattern Generator: $P_N : I_{peak}(\forall I_{sample}(P_N, SRMS)), \ I_{averaged}(P_N, SRMS)$
$$Fixness(P_N) = I_{Measurement}(P_N, T) \in \{I_{peak} \quad I_{averaged}\}$$

}

$$Averaged\_Fixness(Fixness(P_N^{POP})) = \frac{\sum_{N=0}^{N} I_{Measurement}(P_N)}{N}, N, P_N > 0$$

$if\,(Averaged\_Fixness(I_{Measurement}(P_N^{POP})) > I_{Max\_REF})$
{ Final VCM Generation (Database 1) exit(1) }

For P2 = 0, 1, 2, 3, ..., *Loop1*+1 do :
{

$$Sorting(I_{Measurement}(P_N) \in \{I_{\min}(P_{N_{\min}}) \quad ..... \quad I_{\max}(P_{N_{\max}})\}) \Rightarrow Parent(I_{Measurement}(P_N))$$
$$N \in \{N_{\min} \quad N_{\max}\} \qquad N \in \{N_{\min} = N_{\max} - (N_{\max} \times B) \quad N_{\max}\}$$

$$CrossOver(P_N(C_1, C_2), P_{N+1}(C_3, C_4)) \Rightarrow Upper\_CrossOver(P_N(C_3, C_2), P_{N+1}(C_1, C_4))$$
$$\Rightarrow Lower\_CrossOver(P_N(C_1, C_4), P_{N+1}(C_3, C_2))$$
$$\Rightarrow Stripe\_CrossOver(P_N(C_4, C_3), P_{N+1}(C_2, C_1))$$

$$Recombination(P_N, P_{N+1}) \Rightarrow I_{maximum}(P_N, P_{N+1}) \Rightarrow I_{Best}(P_M),\ N, M, P_N > 0$$

$$Mutation(P_M(C_1, C_2, C_3, C_4 .... C_y)) \Rightarrow P_M(C_1 + R_1, C_2 + R_2, ... C_y + R_y)$$
$$R_y \in \{1 \quad 0 \quad -1\},\ M, P_M, y > 0$$

For P3 = 0, 1, 2, 3, ..., *M*+1 do :
{

$Pattern\_Generator(Vector\_Memory(P_M))$
$\Rightarrow Pattern\_Controller(Vector\_Memory(P_M))\ M > 0$ *(Pattern Executor)*

Start Pattern Generator: $P_M(T_{\min}, T_{\max}) \Rightarrow Dynamic\_Pattern$
Start Current Measurement & Calculation:

$$I_{Measurement}(P_M, T) = \frac{V_{DD}(P_M, T)}{R_{eff}} + \frac{1}{L_{eff}} \int_{T_{\min}}^{T_{\max}} V_{DD}(P_M, T) dT + \Delta I_{CMOS}(P_M, T),\ \forall T, P_M > 0$$
$$T = SRMS(T_{\min}, T_{\max}) \Rightarrow Random\_Float\_Number(T_{\min}, T_{\max})$$
$$T_{\max} \geq T_{\min},\ \forall T_{\min}, T_{\max} > 0$$

Stop Pattern Generator: $P_M : I_{peak}(\forall I_{sample}(P_M, SRMS)),\ I_{averaged}(P_M, SRMS)$
$$Fixness(P_M) = I_{Measurement}(P_M, T) \in \{I_{peak} \quad I_{averaged}\}$$

}

$$Averaged\_Fixness(Fixness(P_M^{POP})) = \frac{\sum_{M=0}^{M} I_{Measurement}(P_M)}{M},\ M, P_M > 0$$

$if\,(Averaged\_Fixness(I_{Measurement}(P_M^{POP})) > I_{Max\_REF})$
{ Worst Case Pattern Found : Final VCM Generation (Database 1) exit(1) }

} *End Of Loop 1*
} *End of Loop 2*

Update So Far Worst Case Pattern Found : Final VCM Generation (Database 1)
End Of D_GA

Claims:

1. A method of generating a set of test patterns for the simulation and/or test of the layout of an integrated circuit, the method comprising the steps of:
(a) generating a set of test patterns on a random basis;
(b) applying the set of test patterns to the integrated circuit using automatic test equipment (ATE);
(c) determining the outputs of the integrated circuit;
(d) processing the outputs to determine whether predetermined test criteria are met;
(e) depending on the determination in step (d), generating a new set of test patterns on the basis of the set of test patterns generated in step (a) using a genetic algorithm.

2. The method of claim 1, comprising:
(f1) repeating steps (b) to (e) until the predetermined test criteria are met.

3. The method of claim 1, comprising:
(f2) repeating steps (b) to (e) for a predetermined number of times or until the predetermined test criteria are met.

4. The method of claim 3, comprising:
generating a new set of test patterns on a random basis, if the predetermined test criteria are not met after the repetition of steps(b) to (e) the predetermined number of times; and
repeating step (f2) on the basis of the new set of test patterns.

5. The method of any of claims 2 to 4, wherein the predetermined test criteria are met if the set of test patterns is associated with an average fitness above a predetermined value.

6. The method of any of claims 2 to 5, wherein step (e) comprises:
combining some or all of the test patterns according to the genetic algortihm to generate the new set of test patterns.

7. The method of claim 6, further comprising:
selecting test patterns out of the set of test patterns according to predetermined selection criteria; and
combining the selected test patterns according to the genetic algortihm to generate the new set of test patterns.

8. The method of claim 7, comprising:
selecting a test pattern if it is associated with a fitness value greater than a reference value.

9. The method of claim 7 or 8, comprising:
(g) selecting a test pattern if it is associated with the highest fitness value of all unselected test patterns.

10. The method of claim 9, comprising:
repeating step (g) until a predetermined percentage of test patterns has been selected.

11. The method of claim 9, wherein step (e) comprises:
(h) arranging the selected test patterns in the order of associated fitness values;

(i) randomly selecting parent test patterns out of the test patterns as arranged in step (h); and
(j) combining the selected parent test patterns.

12. The method of any preceding claim, wherein the genetic algorithm includes crossing over, re-combination, and/or mutation of selected test patterns.

13. The method of any preceding claim, wherein step (a) comprises:
generating a plurality of sets of test patterns, wherein each set of test patterns is included in a test pattern population.

14. The method of claim 13, comprising:
performing the steps of any of the preceding claim for each population.

15. A method of generating a set of input signals for the simulation and/or test of the layout of an integrated circuit, the method comprising the steps of:
(m) generating a plurality of sets of input signals;
(n) applying the plurality of sets of input signals to the integrated circuit using automatic test equipment (ATE);
(o) determining the outputs of the integrated circuit;
(p) processing the outputs to determine whether predetermined test criteria are met;
(q) depending on the determination in step (p), generating a new plurality of sets of input signals on the basis of the plurality of sets of input signals generated in step (m) using a genetic algorithm.

16. The method of claim 15, wherein the input signals are associated with a number of AC/DC parameters.

17. The method of claim 15 or 16, wherein predetermined parameters of each set of input signals vary from those of other sets of input signals.

18. The method of any of claims 15 to 17, comprising:
(r1) repeating steps (n) to (q) until the predetermined test criteria are met.

19. The method of any of claims 15 to 17, comprising:
(r2) repeating steps (n) to (q) for a predetermined number of times or until the predetermined test criteria are met.

20. The method of any of claims 15 to 19, wherein the predetermined test criteria are met if the plurality of sets of input signals is associated with a worst case of operation situation.

21. The method of any of claims 15 to 20, wherein step (q) comprises:
combining some or all of corresponding input signals of different sets of input signals according to the genetic algortihm to generate a new set of input signals.

22. A method of generating test patterns and input signals for the simulation and/or test of the layout of an integrated circuit, the method comprising:
performing the method of any of claims 1 to 14; and
performing the method of any of claims 15 to 21.

23. A data processing system comprising an automatic test equipment (ATE), the data processing system being arranged to perform the method of any preceding claim.

24. A computer program adapted to perform the method of any of the claims 1 to 22 on a computer associated with an automatic test equipment (ATE).

Abstract

Method of generating a test pattern for the simulation and/or test of the layout of an integrated circuit

There is provided a method of generating a test pattern for the simulation and/or test of the layout of an integrated circuit, the method comprising the steps of:
(a) generating a set of test patterns on a random basis;
(b) applying the set of test patterns to the integrated circuit using automatic test equipment (ATE);
(c) determining the outputs of the integrated circuit;
(d) processing the outputs to determine whether predetermined test criteria are met;
(e) depending on the determination in step (d), generating a new set of test patterns on the basis of the set of test patterns employed in step (c) using a genetic algorithm.

Accordingly, the method employs a genetic algorithm (optimization method) to optimize a set of random patterns based on measurements using an ATE. Thereby, a set of worst case noise patterns can be selected automatically.

Fig. 1

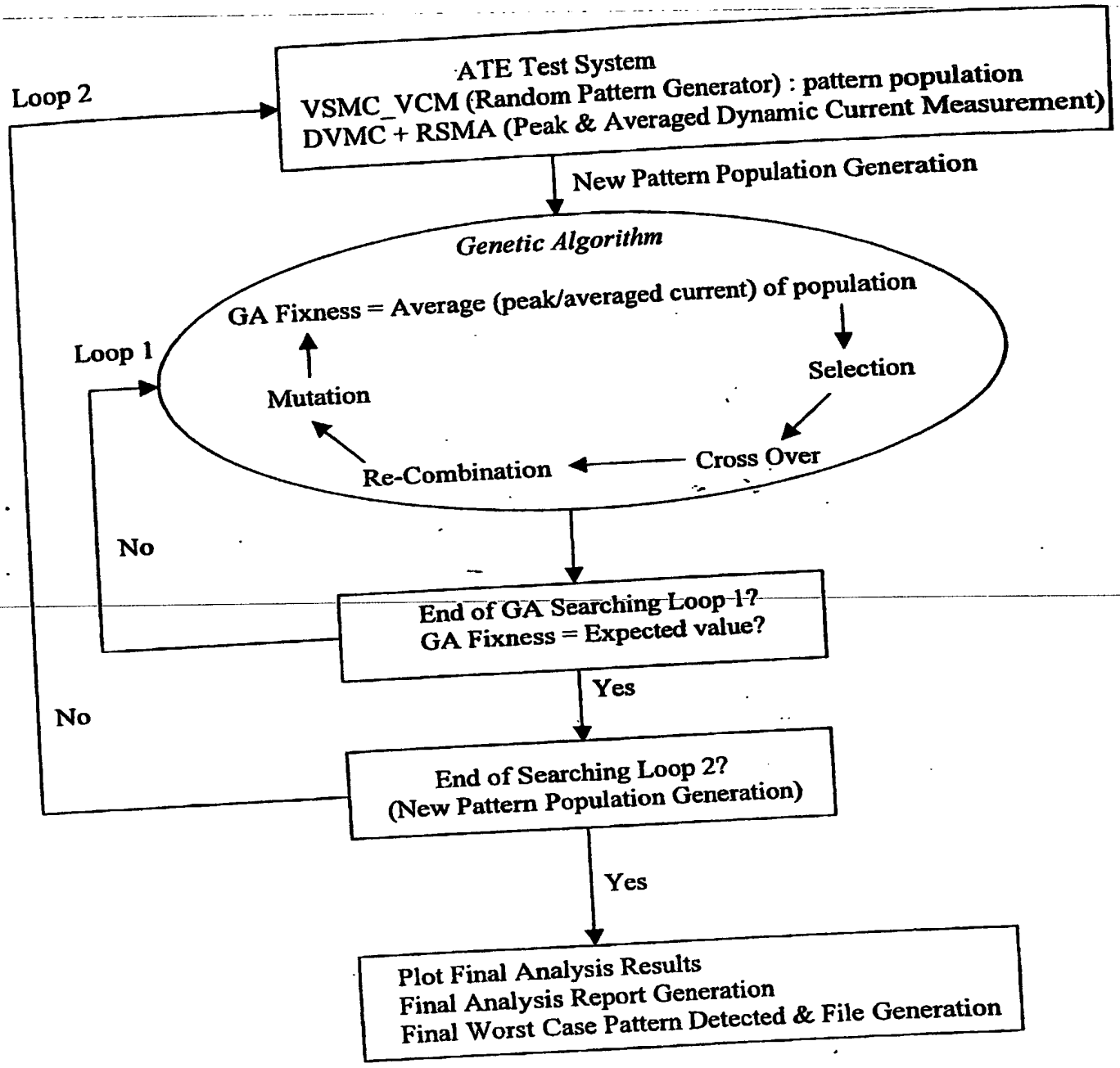


Fig. 1

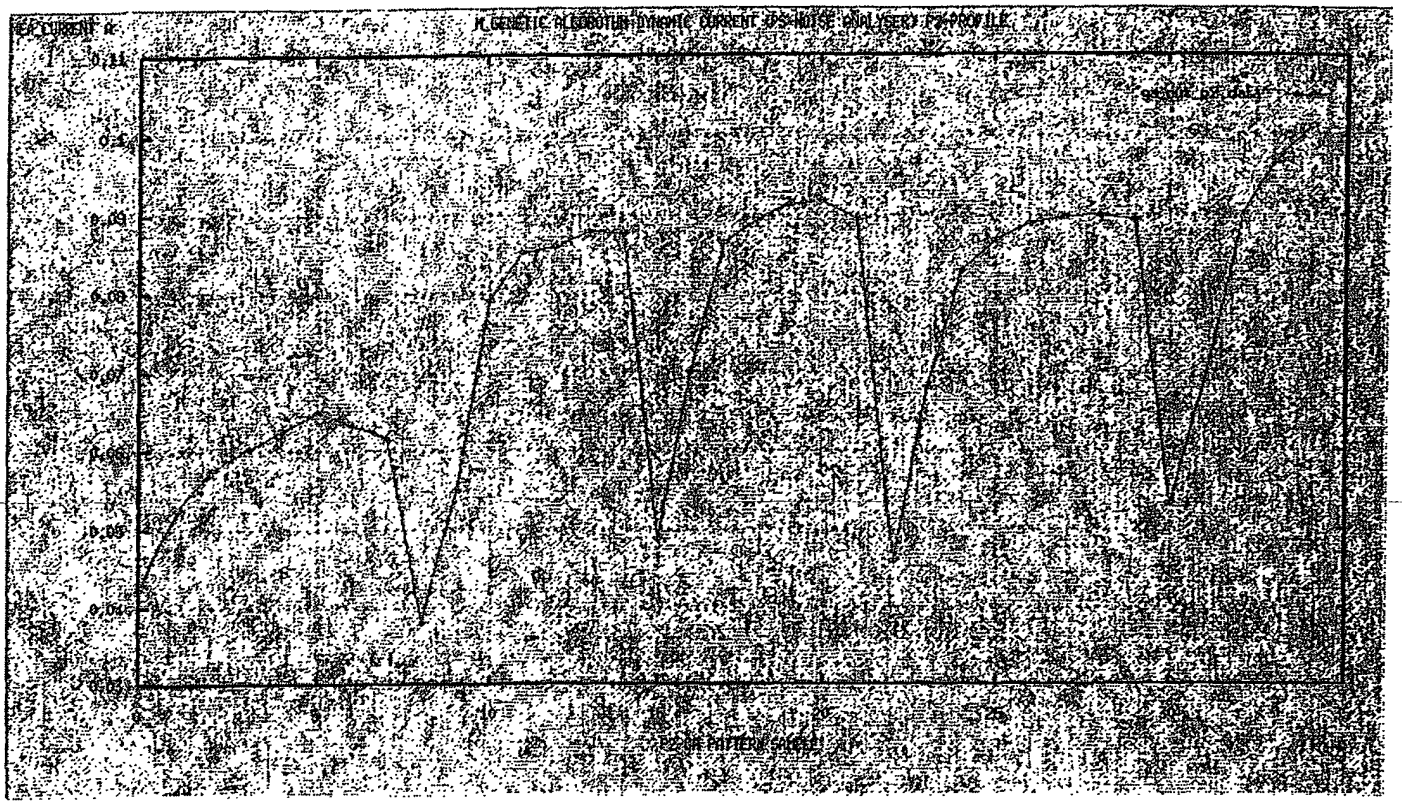
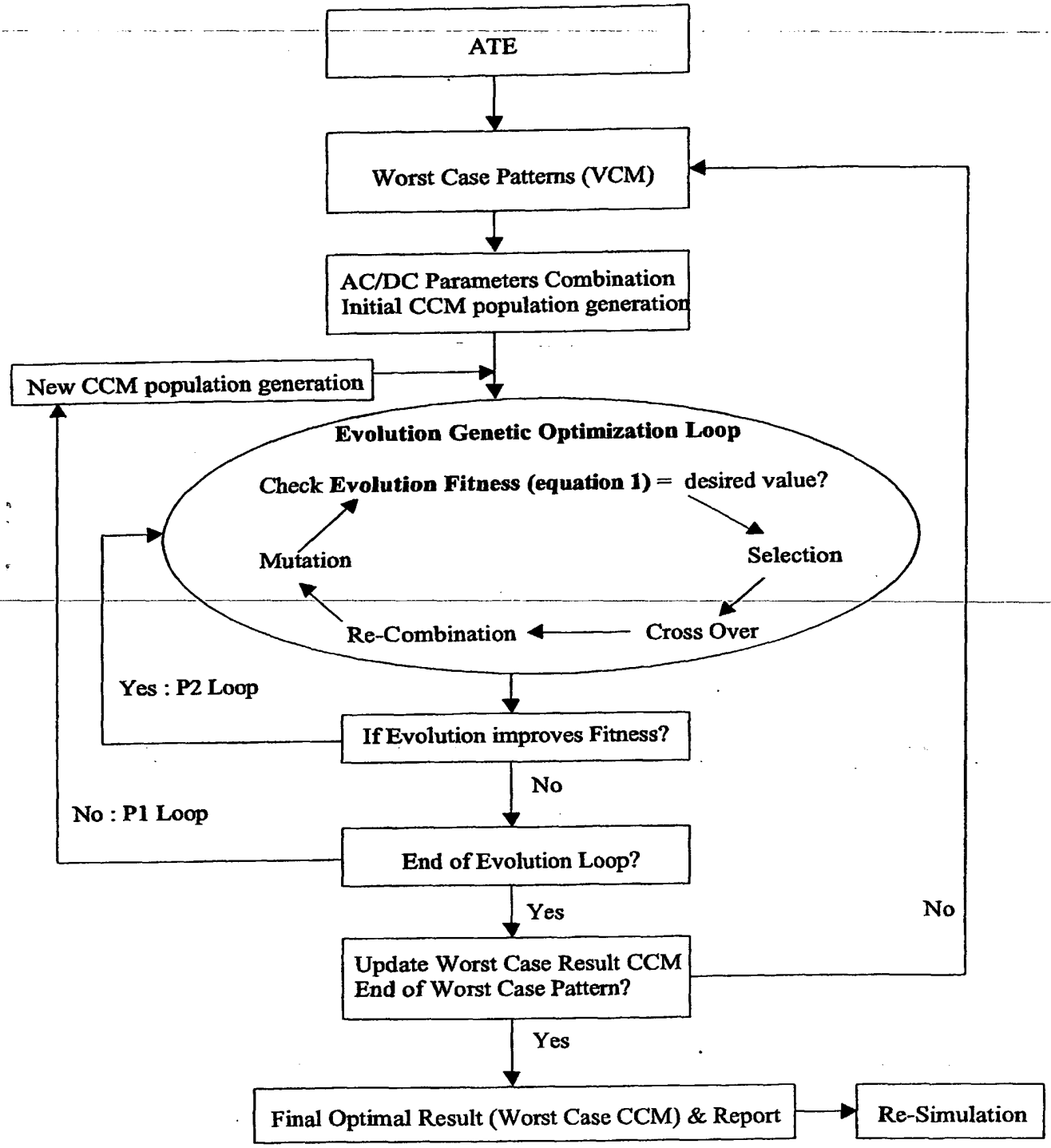

EXCITATION CURRENT
GENETIC ALGORITHM DYNAMIC CURRENT DES NOISE ANALYSED PROFILE
0.11
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
10
20
FOR PATTERN SAMPLE

Fig. 2


ATE
Worst Case Patterns (VCM)
AC/DC Parameters Combination
Initial CCM population generation
New CCM population generation
Evolution Genetic Optimization Loop
Check Evolution Fitness (equation 1) = desired value?
Mutation
Selection
Re-Combination
Cross Over
Yes : P2 Loop
If Evolution improves Fitness?
No
No : P1 Loop
End of Evolution Loop?
Yes
No
Update Worst Case Result CCM
End of Worst Case Pattern?
Yes
Final Optimal Result (Worst Case CCM) & Report
Re-Simulation